To: Examiner Michael Pham, Art Unit 2167, United States Patent and Trademark Office, Fax # 571-273-3924

From: Dan Northfield, for Shimokaji & Associates
Fax: 515-278-0421
Tel: (515) 331-2156

Re: Application No: 10/670,068, SYSTEM AND METHOD FOR DOCUMENT SEARCHING, PROGRAM FOR PERFORMING DOCUMENT-SEARCHING, COMPUTER-READABLE STORAGE MEDIUM STORING THE SAME PROGRAM, COMPILING DEVICE, COMPILING METHOD, PROGRAM FOR PERFORMING THE SAME COMPILING METHOD, COMPUTER-READABLE STORAGE MEDIUM STORING THE SAME PROGRAM, AND A QUERY AUTOMATON EVALUATOR
Final Office Action dated 5/28/08
Faxed pages: 6

Mr. Pham: Attached are the proposed amended claims.
We would like to set up a telephone interview regarding the above case within the next week, with the below proposed agenda, with yourself and your primary examiner.
I, Daniel Northfield, and Carl Giordano, would like to attend the telephone interview for Shimokaji & Associates.

Agenda: 1. Discussion of allowance of proposed amended claims.
2. Discussion of any potential areas of allowance not listed.

1-3 (canceled)

4. (currently amended) A computer-implemented document-searching method for searching a document having a hierarchical structure with elements separated by element identifiers, comprising the steps of:

generating an XPATH query automaton, wherein said query automaton is configured as a table consisting of a state of each node, a type of transition, and a reached state, by storing a query expression input by a compiling device, performing parsing, and identifying different types of nodes in said element identifiers, by the steps of:

generating and registering a state transition by:

replacing an XPATH axis including an XPATH axis in a forward direction that is exemplified as an axis child, or a descendant in an XPath into a state transition,

replacing an XPATH axis including an XPATH axis in an ~~the~~ opposite direction that is exemplified as axis parent, ancestor in an XPath into a state transition,

replacing an XPATH axis including an XPATH axis in a direction of a following-sibling, or a preceding-sibling in said XPath into a state transition,

replacement of a predicate of an XPath into a state transition,

replacement of a logical product (and) of a predicate of an XPath into a state transition,

replacement of a logical add (or) of a predicate of an XPath into a state transition,

replacement of a logical NOT (not) of a predicate of an XPath into a state transition,

and a logical expression including a conjunction ~~or a negative expression~~, while keeping an input query expression equal in terms of search, the query automaton including a plurality of states of the backward nodes, a condition for transition, and at least a search state, wherein said search state includes two states of said input query expression concurrently in a state

transition, and wherein every axis regarding sibling relationship among nodes can be included in the search condition for said query automaton,

said query automaton evaluator determining a state transition of a node under determination by storing a left node and a lower node in correspondence with an identified element identifier, wherein the information obtained from said left node and information obtained from said lower node for a state transition is used concurrently[.]

and evaluating said query automaton with a search result of said left node and said lower node[;],

storing the XPATH query automaton generated by said compiling device in a query automaton storage device; ~~and~~

reading out said XPATH query automaton from said query automaton storage device and storing said query automaton, while reading in said document and performing a stream search with a query automaton evaluator by using states of a plurality of different types of nodes in said element identifiers included in said document and said query automaton, thereby using two inputs and a search state[, ] ~~and~~ ;

storing the output of the query automaton evaluator in a ~~search result~~ storage device ~~means for~~, and thereafter outputting the stored output of the query automaton evaluator and the output of the searched node;

5. (canceled)

6. (previously presented) The document-searching method according to Claim 4, wherein said step of generating an XPATH query automaton comprises a step of generating an XPATH query automaton with a state transition corresponding to an initial state, a final state, and a search state registered thereon.

7-13 (canceled)

14. (currently amended) A computer-implemented compiling method for generating a query automaton for performing a document search, comprising the steps of:

generating and registering a state transition by:

replacing an XPATH axis including an XPATH axis in a forward direction that is exemplified as an axis child, or descendant in an XPath into a state transition,

replacing an XPATH axis including an XPATH axis in an ~~the~~ opposite direction that is exemplified as an axis parent, or ancestor in an XPath into a state transition,

replacing an XPATH axis including an XPATH axis in a direction of a following-sibling or apreceding-sibling sibling in an XPath into a state transition,

replacement of a predicate of an XPath into a state transition,

replacement of a logical product (and) of a predicate of an XPath into a state transition,

replacement of a logical add (or) of a predicate of an XPath into a state transition,

replacement of a logical NOT (not) of a predicate of an XPath into a state transition,

and a logical expression including a conjunction ~~or a negative expression~~ while keeping an input query expression equal in terms of search, and storing a plurality of states of a backward node in correspondence with said backward node into a query automaton storage device;

generating a query automaton by registering a plurality of states of said backward node, a condition for transition, at least a search state, wherein said search state includes two states of said input query expression concurrently in a state transition, wherein every axis regarding sibling relationship among nodes can be included in the search condition for said query automaton, and a reached state in correspondence with each other in said storage device, performing parsing, and identifying different types of nodes in said element identifiers, by the steps of

generating and registering a state transition by the query automaton including a plurality of states of the backward nodes, a condition for transition, and at least a search state

said query automaton evaluator determining a state transition of a node under determination by storing a left node and a lower node in correspondence with an identified element identifier, wherein the information obtained from said left node and information obtained from said lower node for a state transition is used concurrently, and

evaluating said query automaton with a search result of said left node and said lower node;

storing the output of the query automaton evaluator in a ~~search result~~ storage device ~~means~~, and thereafter outputting the stored output of the query automaton evaluator and the output of the searched node.;

15. (Original) The compiling method according to Claim 14, wherein said compiling method comprises a step of identifying said backward node as a left node or a lower node according to a type of said element identifier, and wherein said plurality of states are states of said left node and said lower node.

16-22 (canceled)